AML Summary

Lukas Diebold

December 15, 2025

Contents

1 Math Preliminaries

Gibbs Distribution The Gibbs distribution is a probability distribution that assigns likelihoods to states based on a cost function, with lower-cost states being more probable. Given a set of states $x \in \mathcal{X}$, a cost function $E(x)$, and an inverse temperature parameter $\beta > 0$, the Gibbs distribution is

$$p(x) = \frac{1}{Z} e^{-\beta E(x)}$$

where the partition function Z ensures normalization

$$Z = \sum_{x \in X} e^{-\beta E(x)}$$

2 Conceptual Foundation

2.1 What is Machine Learning?

"ML is a mathematization of epistemolagy!". In philosophy, it is the science of knowledge, the science of what can be known. This is relevant, because in ML we are interested in systems that produce/generate knowledge.

The goal is then to observe 'reality' and draw conclusions from the observations. This can be seen as a perception-action cycle. Where perception is the result of our observations (typically in a data space $\mathcal{X}$) and the actions are part of a hypothesis space. To go from the data space to the hypothesis class $\mathcal{C}$ we generally use an algorithm A. See Figure 1 for an overview.

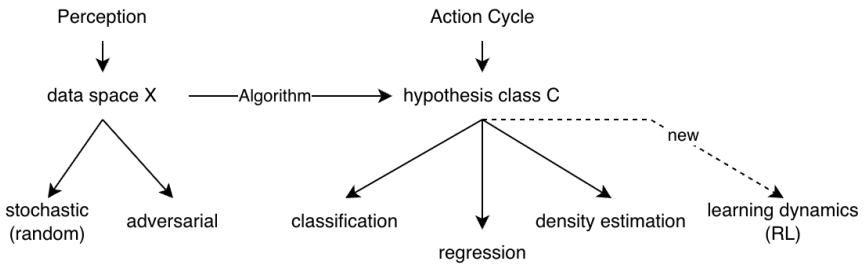

Figure 1: Overview ML

Information processing occurs when $|\mathcal{X}| \gg |\mathcal{C}|$. Taking the example of combinatorial optimization problems, $|\mathcal{X}|$ would be the space of weighted graphs, and we look for a color of the graph or something similar. Then $|\mathcal{X}| \approx K^{\binom{n}{2}}$ and $|\mathcal{C}| \approx e^{n \log n}$ so we obserge that $|\mathcal{X}|$ is much larger.

2.2 Conceptual foundation of inference

1. Perception of realits is mediated by data of senses/sensors
2. Data are stochastic $\rightarrow$ porbabilistic
3. Sensing restricts us to selected aspects of reality
4. Humans interpret data by a huge reduction in degrees of freedom
   $|x| \gg |e|$ (space of graphs $\gg$ space of colorings, cycles, spanning trees)
5. Tuple ({data }, {hypotheses }) define models:

$$\begin{aligned} \mathcal{A} : \mathcal{X} &\longrightarrow \mathcal{C} \\ x &\longmapsto c = \mathcal{A}(x) \end{aligned}$$

6. Experiments ϵ provide us with data
7. Learning means interpreting X w.r.t hypotheses $\mathcal{C}$

2.3 Artificial Intelligence

Taking a high level view. We live in very high dimensional data, which we are not able to fully process. We use algorithms to make sense of the data, and this then informs our values, from this we get the following relation

$$\text{Data} \longrightarrow \text{ Algorithms } \mathcal{A} \longrightarrow \text{ Values}$$

In epistemology, we differentiate between **deduction** and **induction**. Deduction is a form of reasoning in which the conclusion follows necessarily from the premises, while induction tries to generalize, that is, the conclusion goes beyond the premises and generally probabilistic. We can construct a model in which deduction and induction form a **feedback loop**, not two isolated

methods. In simpler terms, on one side we try to formulate axioms from our observations (empirical data), and on the other side we then use these axioms and derive logical consequences from them. This is not anything new, and this cycle generally informs the model of "Theory, Experiment, Computation" in science. What has changed with ML/AI is that we're now in the era of non-parametric modeling.

2.4 Extracting Value from Data - What is the problem?

- Algorithms that process inputs with noise compute random variables as outputs!
- Algorithms should compute typical solutions!
- When do algorithms generalize over noise/model mismatch?
- How can algorithms autonomously improve performance?

2.5 What does Generalization mean?

- Out-of-sample risk

$$\theta^{\star}\left(X'\right) \sim \mathbb{P}^{\mathcal{A}}(\theta \mid X) \in \arg\min_{\mathbb{P}(.|.)} \mathbb{E}_{X'}\mathbb{E}_{\theta|X'}\mathbb{E}_{X''|X'}R\left(\theta, X''\right)$$

  where X' is the training data and X'' is the test data, so thiw is the risk where θ is conditioned on the training data (trained the model).

- Log loss of posterior (risks and probabilities are dependent!)

$$\begin{aligned}\theta^{\star}\left(X'\right) \sim \mathbb{P}^{\mathcal{A}}(\theta \mid X) &\in \arg\min_{\mathbb{P}(.|.)} \mathbb{E}_{X'}\mathbb{E}_{\theta|X'}\mathbb{E}_{X''|X'}\left(-\log \mathbb{P}\left(\theta \mid X''\right)\right) \\ &\in \arg\min_{\mathbb{P}(.|.)} \mathbb{E}_{X'}\mathbb{E}_{\theta|X'}\mathbb{E}_{X''|X'}\left(\beta R\left(\theta, X''\right) + \log Z\right)\end{aligned}$$

- Posterior agreement

$$\theta^{\star}\left(X'\right) \sim \mathbb{P}^{\mathcal{A}}(\theta \mid X) \in \arg\min_{\mathbb{P}(.|.)} \mathbb{E}_{X'}\mathbb{E}_{X''|X'}\left(-\log \mathbb{E}_{\theta|X'}\mathbb{P}\left(\theta \mid X''\right)\right)$$

2.6 Conceptional Foundation of Inference

- Our perception of "our world" / reality is mediated by data of senses / sensors.
- Our data are influenced by chance.
- Creatures interpret selected aspects of reality by hypotheses to "survive and reproduce"
- Data and hypotheses define models to enable judgements, decisions and actions.
- AI / ML: Algorithms define relations of data and hypotheses, e.g., they select models !

3 Fundamentals of Machine Learing

Bayes Rule:

$$\mathbb{P}(\text{model} \mid \text{data}) = \frac{\mathbb{P}(\text{ data } \mid \text{ model })\mathbb{P}(\text{ model })}{\mathbb{P}(\text{ data })}$$

ML method:

$$\widehat{\text{model}}_m \in \arg\max_{\text{model}} \mathbb{P}(\text{data} \mid \text{model})$$

where $\widehat{\text{model}}_n$ is consistent, asymptotically normal, and asymptotically efficient

consistency: a point estimated $\hat{\theta}_n$ of the parameter $\theta = \theta_0$ is consistent if

$$\forall \varepsilon > 0, \mathbb{P}\left(\left|\hat{\theta}_n - \theta_0\right| > \varepsilon\right) \xrightarrow{n\to\infty} 0$$

more formally

$$\forall \theta, \forall \varepsilon, \delta > 0, \exists n_0, \forall n > n_0, \mathbb{P}\left(\left|\widehat{\theta}_n - \theta\right| < \varepsilon\right) > 1 - \delta$$

efficiency:

$$\hat{\theta}_n = \arg\operatorname{uin}_{\hat{\theta}} \mathbb{E}\left[\left(\hat{\theta}_n - \theta_0\right)^2\right]$$

3.1 Efficiency: Rao Cramer bound

Problem: Precision of parameters estimation.
Given the likelihood $p(y \mid \theta)$ for $\theta \in \Theta$, data $y_1, \ldots, y_n \sim p(y \mid \theta = \theta_0)$ we are interested in the question: "How precisely can we estimate $\theta = \theta_0$ given n samples?". To answer this we define an estimator $\hat{\theta}(y_1, \ldots, y_n)$ and estimate the expected deviation

$$\mathbb{E}_{y|\theta}\left[(\hat{\theta} - \theta)^2\right]$$

Score: $\Lambda = \frac{\partial}{\partial\theta} \log p(y \mid \theta) = \frac{\frac{\partial}{\partial\theta} p(y|\theta)}{p(y|\theta)}$
bias : $b_{\hat{\theta}} = \mathbb{E}_{y|\theta}\left[\hat{\theta}(y_1, \ldots, y_n)\right] - \theta$
Expected score :

$$\begin{aligned}\mathbb{E}_{y|\theta}[\Lambda] &= \int p(y \mid \theta) \frac{\frac{\partial}{\partial\theta} p(y \mid \theta)}{p(y \mid \theta)} dy \\ &= \frac{\partial}{\partial\theta} \int p(y \mid \theta) dy = \frac{\partial}{\partial\theta} 1 = 0\end{aligned}$$

Expected score times estimator product :

$$\begin{aligned}\mathbb{E}_{y|\theta}[\Lambda\hat{\theta}] &= \int \rho(y \mid \theta) \frac{\frac{\partial}{\partial\theta} \rho(y \mid \theta)}{\rho(y \mid \theta)} \hat{\theta} dy \\ &= \frac{\partial}{\partial\theta}\left(\int \rho(y \mid \theta)\hat{\theta} dy - \theta\right) + 1 \\ &= \frac{\partial}{\partial\theta}\left(\mathbb{E}_{y|\theta}\hat{\theta} - \theta\right) + 1 = \frac{\partial}{\partial\theta} b_{\hat{\theta}} + 1\end{aligned}$$

Cross correlation between score and estimator :

$$\mathbb{E}_{y|\theta}[(\Lambda - \underbrace{\mathbb{E}\Lambda}_{=0})(\hat{\theta} - \mathbb{E}\hat{\theta})] = \mathbb{E}_{y|\theta}[\Lambda\hat{\theta}] - \underbrace{\mathbb{E}_{y|\theta}[\Lambda]}_{=0} \mathbb{E}\hat{\theta}$$

Cauchy-Schwarz inequality :

$$\begin{aligned}
&\left(\mathbb{E}_{y|\theta}[\Lambda(\hat{\theta}-\mathbb{E}\hat{\theta})]\right)^2 \leq \mathbb{E}_{y|\theta}\left[\Lambda^2\right]\mathbb{E}_{y|\theta}\left[(\hat{\theta}-\theta-\mathbb{E}\hat{\theta}+\theta)^2\right] \\
&= \mathbb{E}_{y|\theta}\left[\Lambda^2\right]\left(\mathbb{E}_{y|\theta}\left[(\hat{\theta}-\theta)^2\right]-\mathcal{L}(\mathbb{E}\hat{\theta}-\theta)^2+(\mathbb{E}\hat{\theta}-\theta)^2\right) \\
&= \mathbb{E}_{y|\theta}\left[\Lambda^2\right]\left(\mathbb{E}_{y|\theta}\left[(\hat{\theta}-\theta)^2\right]-b_{\hat{\theta}}^2\right) \\
&\mathbb{E}_{y|\theta}\left[(\hat{\theta}-\theta)^2\right] \geq \frac{\left(\mathbb{E}_{y|\theta}[\Lambda\hat{\theta}]\right)^2}{\mathbb{E}_{y|\theta}\left[\Lambda^2\right]}+b_{\hat{\theta}}^2 = \frac{\left(\frac{\partial}{\partial\theta}b_{\hat{\theta}}+1\right)^2}{\mathbb{E}_{y|\theta}\left[\Lambda^2\right]}+b_{\hat{\theta}}^2
\end{aligned}$$

-¿ General Rao Cramer bound for estimator $\hat{\theta}$

$$\mathbb{E}_{y|\theta}\left[(\hat{\theta}-\theta)^2\right] \geq \frac{\left(\frac{\partial}{\partial\theta}b_{\hat{\theta}}+1\right)^2}{\mathbb{E}_{y|\theta}\left[\Lambda^2\right]}+b_{\hat{\theta}}^2$$

Fisher information : $\mathbb{E}_{y|\theta}\left[\Lambda^2\right] = \int \rho(y \mid \theta)\left(\frac{\partial}{\partial\theta}\log\rho(y \mid \theta)\right)^2 dy =: I(\theta)$
Remartes
II) Note tradeoff $\frac{\partial}{\partial\theta}b_{\hat{\theta}} < 0$ vs. $b_{\hat{\theta}}^2$ for biased estimators! unbiased estimators unght not be the best estimators!
case with $\boldsymbol{n}$ samples

$$\begin{aligned}
\mathbb{E}_{y_1,\ldots,y_n|\theta}\left[\Lambda^2\right] &= \int p\left(y_1,\ldots,y_n \mid \theta\right) \underbrace{\frac{\partial}{\partial\theta}\log p\left(y_1,\ldots,y_n \mid \theta\right)}_{=\frac{\partial}{\partial\theta}\sum_{i=n}\log p(y_i|\theta)=\sum_{i=n}\Lambda_i})^2 dy_1\ldots dy_n =: I^{(n)}(\theta) \\
&= \int p\left(y_1,\ldots,y_n \mid \theta\right)(\sum_{i=n}\Lambda_i^2+\underbrace{\sum_{i=n}\sum_{j=n}\Lambda_i\Lambda_j}_{=0})dy_1\ldots dy_n \\
&= \sum_{i=n}\int p\left(y_i \mid \theta\right)\Lambda_i^2 dy_i = nI(\theta)
\end{aligned}$$

Remark: The Fisher information of n iid. r.v. is $n\times$ Fishes information of 1 r.v.

4 Regression

4.1 Act 1: High Dimensional regression is unstable

we assume X and y are distributed according to a distribution p_* (i.e. $X, y \sim p_*$) and

$$y = f_*(x) + \varepsilon \quad \text{with } \varepsilon \sim \mathcal{N}(0, \sigma)$$

Then our task ist to estimage f_* from $D = \{x_i, y_i\} \sim p_*$. The problem in this form isn't tractable, this is why we restrict the choice of functions. First we choose the limit ourselfs to linear functions, that is function of the form

$$f_*(x) = \beta^\top x$$

This problem we then solve using Maximum Likelihood Estimation (MLE), that is

$$\begin{aligned}
\widehat{\beta} &= \arg\max_{\beta \in \mathbb{R}^R} p(D \mid \beta) \\
&= ... \\
&= \arg\min \mathrm{MSE}(D, \beta) \\
&= \frac{1}{n} \sum_{i \leq n} (y_i - \beta^\top x_i)^2 \\
&= (X^\top X)^{-1} X^\top y \\
&= ... \\
&= X^\top (X X^\top)^{-1} y
\end{aligned}$$

which is known as the ordinary least squares estimator, where

$$X = \begin{bmatrix} -x_1- \\ \vdots \\ -x_n- \end{bmatrix} \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

This estimator has some interesting properties. It is unbiased and has the minimal variance for all unbiased estimators (Gauss-Markov Theorem). Thus, from the formula

$$\text{error} = \text{bias}^2 + \text{variance} + \text{noise}$$

we find that this estimator is the one with the smallest error of all the unbiased estimators. Then why does no-one use this estimator, this is because if we introduce a bit of bias, we can significantly reduce the variance. So what is the variance of $\mathrm{Var}(\hat{b})$? For this we use the SVD of $X = UDV^\top$ if we plug this into the formula from above we get

$$\hat{b} = VD^{-1}U^\top y$$

so, since y has a gaussian distribution and we just multiply it with a matrix $(VD^{-1}U^\top)$, $\hat{b}$ also has a gaussian distribution. Then

$$\mathrm{Var}(\hat{\beta}) = \ldots = \sigma^2 V D^{-2} V^\top = \sigma^2 \sum_{i \leq n} \frac{1}{D_{ii}^2} V_i V_i^\top$$

What are the implications of this? If we are working with high dimensional data, many of these features are typically correlated, this means that X has a low rank, then the smallest D_{ii} values are very small, which means that the bias is huge.

4.2 Act 2: Stability via Regularization

The typical process of **Bayesian inference** goes through the follwing stages:

1. Prior $\beta \sim \mathcal{N}(0, \tau^2 I)$
2. Observe likelihood $p(D|\beta)$

3. Posterior *(adjustment of the likelihood)* $p(\beta|D) \propto p(\beta)p(D|\beta) \propto \exp\left(-\frac{1}{2\sigma^2}\text{MSE}(D,\beta) - \frac{1}{2\tau^2}\|\beta\|^2\right)$

Observe that the last term is equivalent to the loss induced by the **Ridge**-Regression. (If we change β to have a Laplace distribution (heavy tails), we get **Lasso**-Regression) By adjusting τ we can adjust the bias-variance tradeoff. If we do the math we get

$$\hat{\beta}_{\text{MAP}} = \left(x^\top x + \frac{\sigma^2}{\tau^2} I\right)^{-1} x^\top y$$

and if we do again an SVD to calculate the variance we get

$$\text{Var}(\hat{\beta}_{\text{MAP}}) = \sigma^2 \sum_{i \leq n} \frac{D_{ii}^2}{\left(D_i^2 + \frac{\sigma_i^2}{\tau^2}\right)} V_i V_i^\top$$

So we can use τ to control the whole fraction term, which is an improvement to the situation we had before.

4.3 Act 3: Polynomial regression via

Now we change our assumption for $f_*(x)$ so that it is a polynomial function on x. We have

$$f_*(x) = \varphi(x)^\top \beta_*$$

where $\beta_* \in \mathbb{R}^\infty$ and

$$\varphi(X) = K_x \left(\frac{x_1^{\alpha_1} \ldots x_d^{\alpha_d}}{\sqrt{\alpha_1! \ldots \alpha_d!}}\right)_{\alpha \in \mathbb{N}^d}$$

Then for $x, x' \in \mathbb{R}^a$

$$\begin{aligned}\varphi(x)^\top \varphi(x') &= K_{RBF}(x, x') \\ &= \exp\left(-\frac{1}{2}||x - x'||^2\right)\end{aligned}$$

and

$$\begin{aligned}\hat{\beta} &= \arg\min_{\beta \in \mathbb{R}^\infty} \frac{1}{n} \sum_{i \leq n} \left(y_i - \varphi(x_i)^\top \beta\right)^2 \\ &= \Phi^\top \left(\Phi\Phi^\top\right)^{-1} y\end{aligned}$$

where

$$\Phi = \begin{bmatrix} \varphi(x)^\top \\ \varphi(x_2)^\top \\ \varphi(x_n)^\top \end{bmatrix} \in \mathbb{R}^{n \times \infty}$$

This doesn't look good. But let x_* be a test point. Then

$$\begin{aligned}x_x &= \varphi(x_*)^\top \hat{\beta} \\ &= \varphi(x_*)^\top \Phi^\top \left(\Phi\Phi^\top\right)^{-1} \\ &= k(x_*)K\end{aligned}$$

where $k(x_*) = \left(\varphi(x_*)^\top \varphi(x_i)\right)_{1 \leq n}$ and $K_{ij} = \left(\varphi(x_i)^\top \varphi(x_j)\right)$

The problem is that the inversion of the matrix is $O(n^3)$, which is a problem if our data are too high-dimensional.

4.4 Act 4: Neural Networks

We assume f_* has only a single, very wide hidden layer.

$$f_*(X) = \frac{1}{\sqrt{m}} \sum_{i \leq m} \alpha_i \phi\left(\omega_i^\top X\right)$$

then $\theta = \{\alpha_i, w_i\}_{i \leq m}$. We use gradient descent. We initialize our NN with

$$\theta_0 \sim N\left(0, w^2\right])$$

and we update our parameters using gradient descent.

$$\theta_{t+1} \leftarrow \theta_t - \eta \nabla_0 \text{ MSE(D, } \theta_t)$$

and we can calculate

$$\nabla_\theta \operatorname{MSE}\left(D, \theta_t\right) = \widetilde{\Phi}_t^\top \left(f_t - y\right)$$

where

$$\tilde{\Phi}_t = \left(-\nabla_0 f\left(x; \theta_t\right)^\top -\right)_{i \leq n} \text{ and } f_t = \left(f(x; \theta_t)\right)_{i \in n}$$

If we approximate f_t using a first order taylor approximation

$$f_t \approx t_0 + \tilde{\Phi}\left(\theta_t - \theta_0\right)$$

then

$$\theta_t - \theta_0 = \Phi^\top \left(\tilde{\Phi}\tilde{\Phi}^\top\right)^{-1} \left(f_t - f_0\right)$$

Then let x_* be a test point

$$f_t\left(x_t\right) \approx f_0\left(x_k\right) + \nabla f\left(x_t, \theta_t\right)^\top \tilde{\Phi}_t{}^\top \left(\Phi_t \tilde{\Phi}_t{}^\top\right)^{-1} \left(f_t - f_0\right)$$

if we then let t and m go ∞, this turns into

$$f_t\left(x_t\right) \ = 0 + \widetilde{K}\left(x_*\right)^\top K^{-1}(y - 0) = k\left(x_*\right)^\top K^{-1} y$$

This is exactly the result we also had at the end of act 3. **So gradient descent with NN is just simulating kernelized ridge-regression, except that we are using a different kernel.**